



# 01. Introducción a la programación

Programación - 1º DAM

Luis del Moral Martínez

versión 20.10

Bajo licencia CC BY-NC-SA 4.0



# Contenidos del tema

1. Concepto de computadora
2. Estructura de una computadora
3. Representación de la información
4. Concepto de algoritmo
5. Paradigmas de programación
6. Programación estructurada
7. Programación orientada a objetos
8. Tipos de software
9. Lenguajes de programación

# 1. Concepto de computadora

## ¿Qué es una computadora?

- Es un *dispositivo electrónico*
- Se encarga de procesar **datos** e **información** para obtener **resultados**
- Los datos y la información se introducen por la **entrada** (input)
- Los resultados se obtienen por la **salida** (output).
- Está formada por **componentes físicos** (hardware)
- Las instrucciones que gobiernan a la computadora se denominan **programas** (software)
- La persona que se encarga de desarrollar programas se denomina **programador**

## 2. Estructura de una computadora

### Componentes de una computadora

1. **Unidad Central de Proceso (CPU):** controla el proceso de información de la computadora
  - **Unidad Aritmética y Lógica (ALU):** realiza operaciones aritméticas y lógicas
  - **Unidad de Control (UC):** coordina las operaciones de la computadora y determina la ejecución
  - **Registros:** memoria muy rápida que permite almacenar información temporal para su uso en la CPU
2. **Memoria principal (RAM):**
  - Almacena instrucciones (partes de un programa) y datos (usados por los programas)
  - Para que un programa se ejecute debe ubicarse en la memoria principal
  - Permite el acceso *aleatorio*, pero cuando la computadora se apaga, se borra su contenido

## 2. Estructura de una computadora

### Componentes de una computadora

#### 3. Memoria de solo lectura (ROM):

- Su contenido es permanente (viene pregrabada por el fabricante, aunque puede actualizarse)
- El sistema operativo la utiliza para inicializar la computadora

#### 4. Memoria externa (almacenamiento permanente):

- Constituye el almacenamiento masivo (Discos mecánicos, discos SSD, CD-ROM, DVD-ROM, USB)
- Son dispositivos de acceso secuencial o aleatorio de carácter permanente
- Se utilizan para almacenar programas (entre los que se incluye el sistema operativo) y datos

# 2. Estructura de una computadora

## Componentes de una computadora

### 5. Programa:

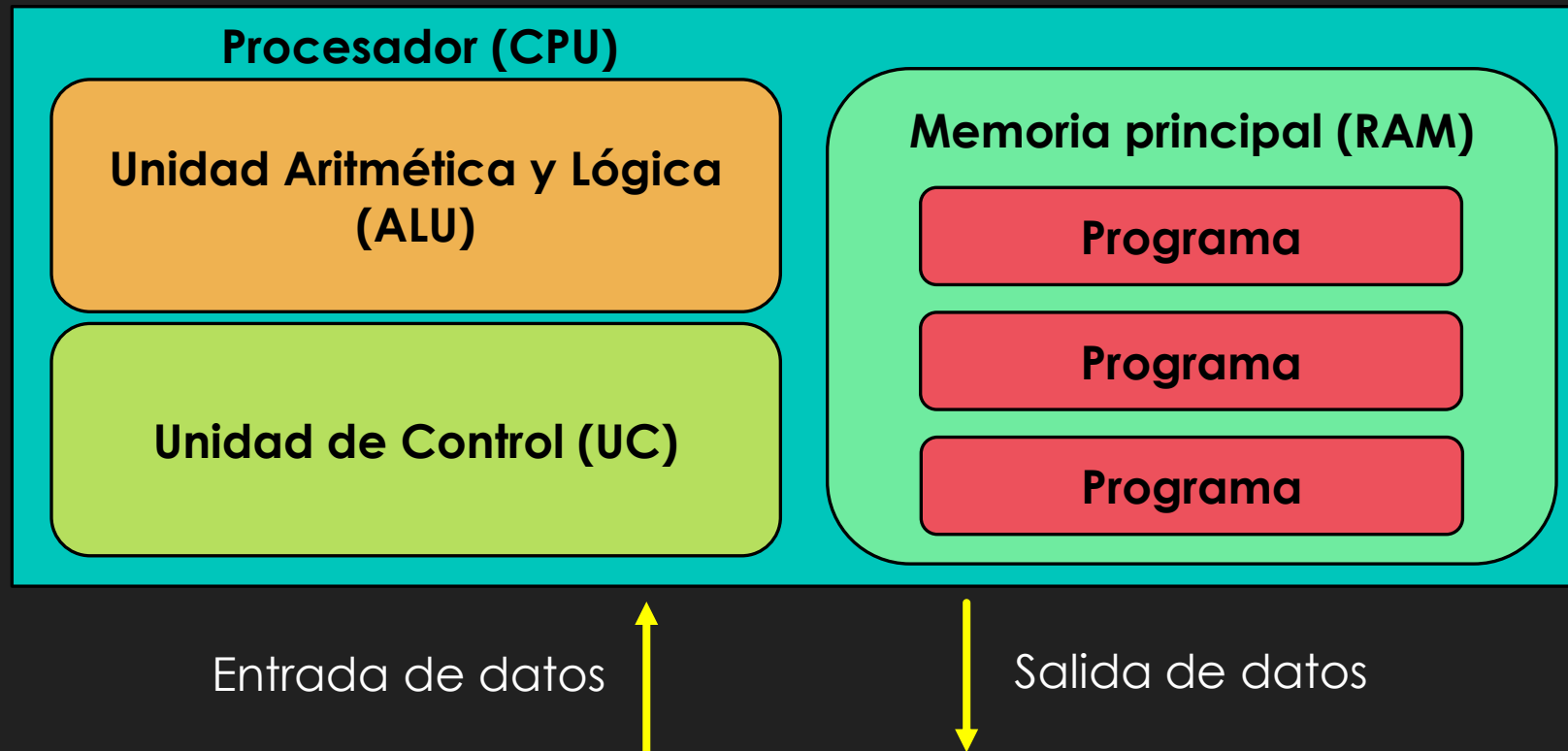
- Conjunto de instrucciones que puede ejecutarse en una computadora
- Las instrucciones se ejecutan en un determinado orden
- El programa hace uso de los componentes de la computadora para lograr su objetivo
- El programa se debe transferir a la memoria RAM para poder ser ejecutado en la CPU

### 6. Dispositivos de entrada/salida (E/S):

- Permiten la comunicación entre la computadora y el usuario (teclado, ratón, monitor...)

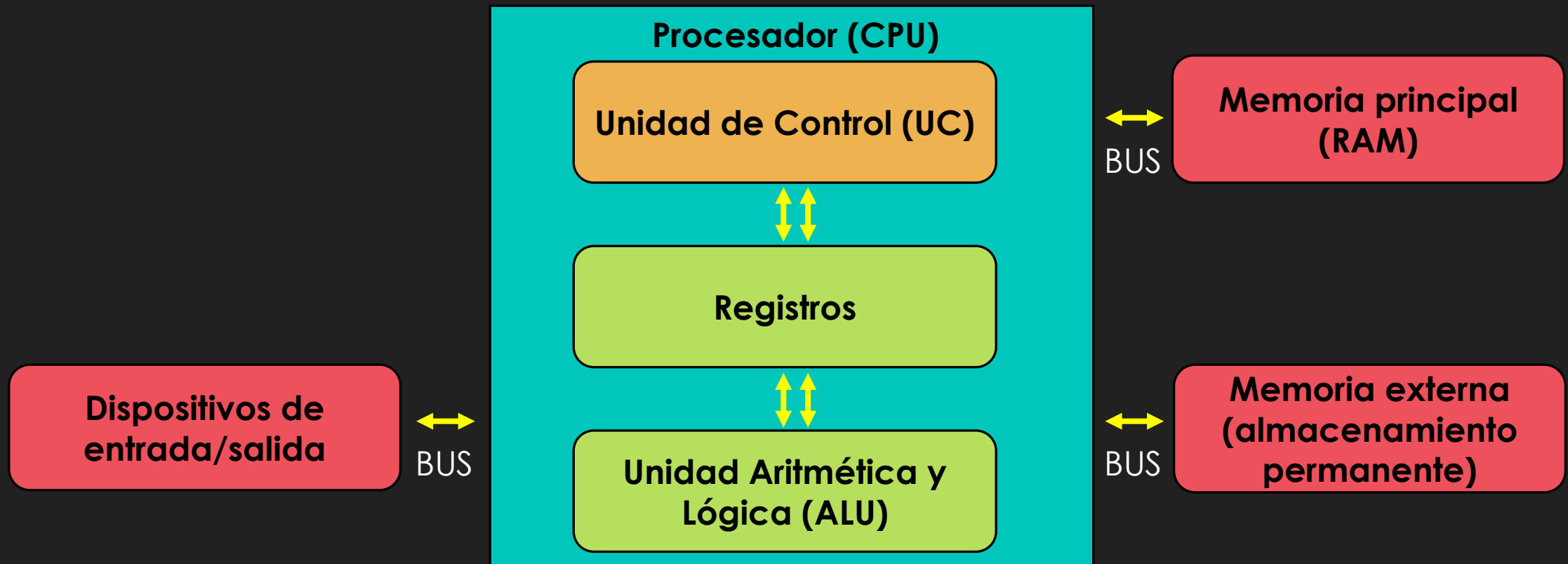
## 2. Estructura de una computadora

### Diagrama de una computadora (1)



## 2. Estructura de una computadora

### Diagrama de una computadora (2)





## 2. Estructura de una computadora

### Organización de la memoria RAM

- La memoria RAM se organiza en *celdas de almacenamiento*
- Las celdas, o posiciones, se denominan también **palabras**
- El número de celdas consta a su vez de un **número de bits**
- Un **bit** puede almacenar o un 0 o un 1
- Los bits se organizan en **bytes**. 1 byte equivale a 8 bits
- Cada celda de almacenamiento tiene una **dirección** y un **contenido**
- Cuando se almacena una nueva información en una posición se destruye el contenido anterior

## 2. Estructura de una computadora

### Diagrama de organización de la memoria RAM

Dirección de  
memoria

999

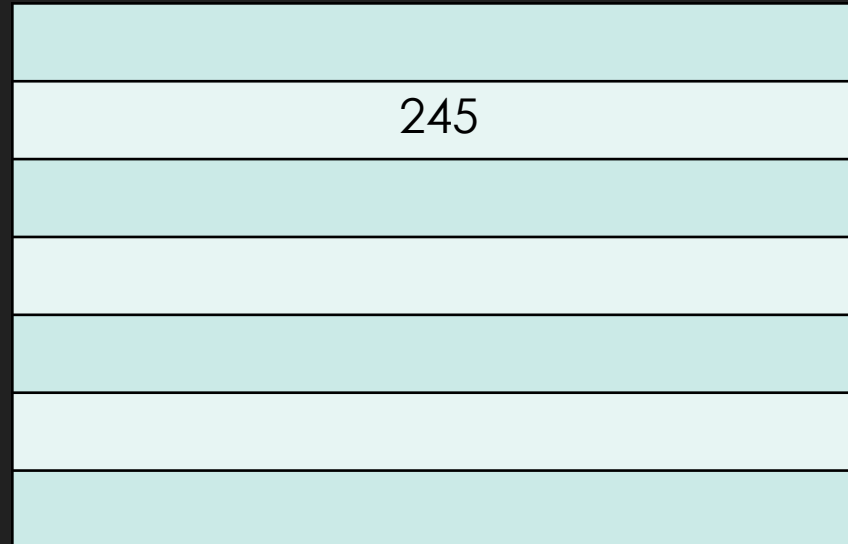
998

...

2

1

0



Contenido de la  
dirección 998

## 2. Estructura de una computadora

### Unidades de medida de almacenamiento

- Estas unidades se aplican para los registros, la ROM, la RAM y el almacenamiento permanente

Unidad	Equivalencia
Byte (B)	8 bits
Kilobyte (KB)	1024 bytes
Megabyte (MB)	1024 Kbytes
Gigabyte (GB)	1024 Mbytes
Terabyte (TB)	1024 Gbytes
Petabyte (PB)	1024 Tbytes
1 TB = 1024 GB = 1024 x 1024 Mb = 1.048.576 KB = 1.073.741.824 Bytes	

# 3. Representación de la información

## ¿Cómo se representa la información?

- Toda la información se almacena en forma de bits en la memoria (RAM, ROM o externa):
  - **Caracteres alfabéticos:** A, B, C...
  - **Caracteres numéricos:** 0, 1, 2...
  - **Caracteres especiales:** {, }, >, !, #, \$...
  - **Caracteres de control:** retorno de carro (para saltar de línea)...
- El texto que se introduce en una computadora se codifica usando un código
- El código asigna a cada carácter una determinada combinación de bits

# 3. Representación de la información

## Codificación de la información

- **Código EBCDIC** [\[más información\]](#):

- Utiliza 8 bits (máximo 256 símbolos diferentes)
- Usado al inicio por IBM

- **Código ASCII** [\[más información\]](#):

- Utiliza 7 bits (máximo 128 caracteres diferentes).
- También existe el ASCII extendido (8 bits), que es el más utilizado

- **Código Unicode** [\[más información\]](#):

- Utiliza 16 bits (hasta 65536 patrones diferentes)
- Aporta soporte para otras lenguas (Español, Chino, Japonés, Árabe, etcétera)

Código	Valor	Carácter
EBCDIC	11010010	K
ASCII	bin=00101011, dec=43, hex=2B	+
Unicode	U+0025	%

# 4. Concepto de algoritmo

## ¿Qué es un algoritmo?

- Un programador crea un programa para **resolver un problema**
- La técnica que regula esta disciplina se denomina **metodología de la programación**
- El eje central de esta metodología consiste en **desarrollar algoritmos**
- Un **algoritmo** es un *método para resolver un problema*
- Los algoritmos pueden representarse en modo texto (*pseudocódigo*) o gráficamente

# 4. Concepto de algoritmo

## Etapas para resolver un problema

1. Diseñar un algoritmo:
  - Secuencia ordenada de pasos para resolver el problema
2. Traducir el algoritmo a un **lenguaje de programación**:
  - Traducción a un lenguaje de programación
  - El **lenguaje de programación** puede ser ejecutado en una computadora
3. Ejecutar y validar el programa en una computadora
  - El programa se ejecuta y se observan los resultados
  - El programa es corregido si se detecta algún error

# 4. Concepto de algoritmo

## Características de un algoritmo

- Un algoritmo posee tres características clave:
  - Es **preciso**, sin ambigüedades, y consiste en una serie de **instrucciones ordenadas**
  - Es **definido**, de forma que si se sigue el algoritmo dos veces, se debe obtener el mismo resultado
  - Es **finito**, de manera que el algoritmo tiene un **inicio** y un **final**



# 4. Concepto de algoritmo

## Pseudocódigo

- Representación textual de un algoritmo
- Se emplean verbos (en infinitivo) y nombres
- Se utiliza un idioma fácilmente comprensible

### Ejemplo: algoritmo que busca un número primo N

1. Inicio
2. Sea  $x = 2$
3. Sea  $r = N/x$  (división de N entre x)
4. Si r es entero, N es primo (ir al paso 7), en caso contrario continuar en el paso 5
5. Sea  $x = x + 1$
6. Si x es igual a N, N es primo (ir al paso 7), en caso contrario saltar al paso 3
7. Fin






# 4. Concepto de algoritmo

## Representación gráfica de un algoritmo

- Los algoritmos pueden representarse utilizando símbolos
- Se sigue el estándar ISO 5807 (1985)

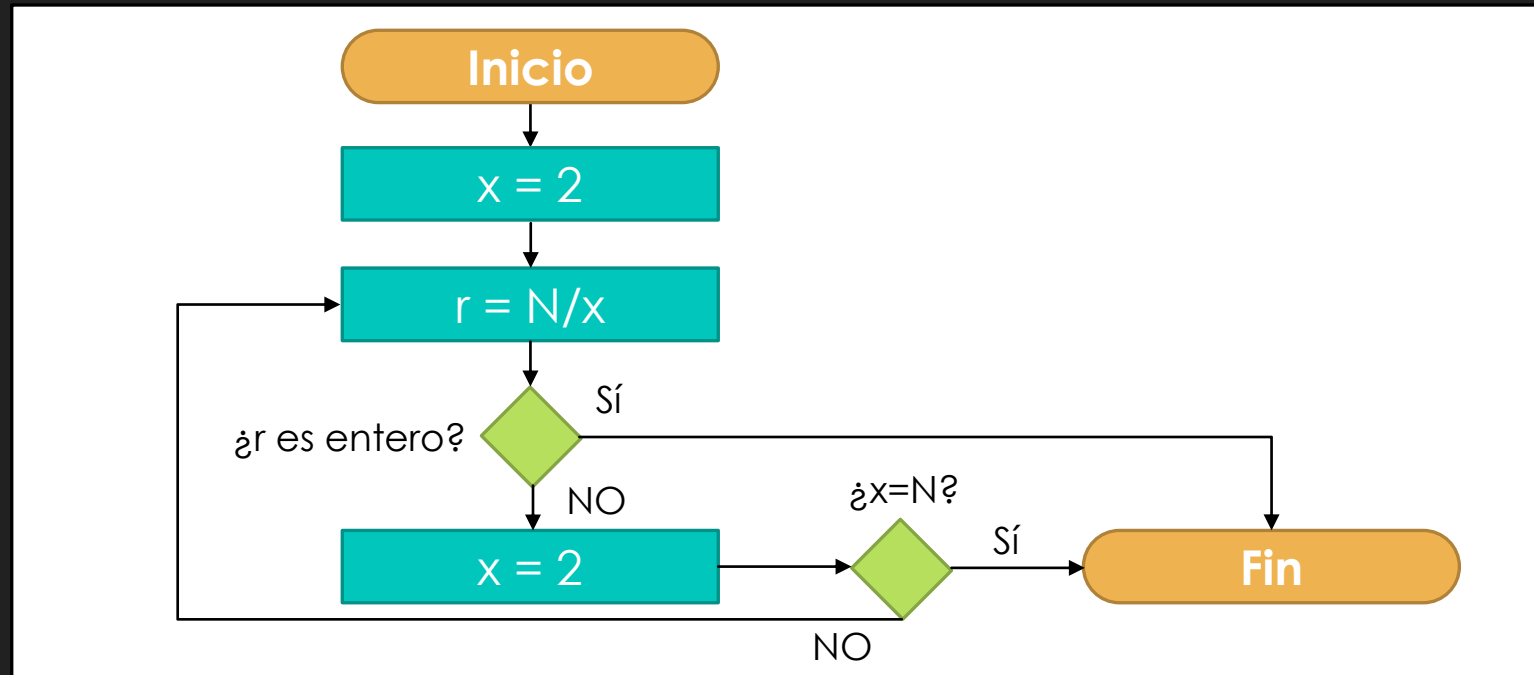
# 4. Concepto de algoritmo

## Representación gráfica de un algoritmo

Símbolo	Significado
	Flujo del algoritmo
	Inicio y fin del algoritmo
	Proceso
	Decisión
	Entrada/salida

# 4. Concepto de algoritmo

## Ejemplo de un diagrama de un algoritmo



# 5. Paradigmas de programación

## Concepto de paradigma de programación

- Prácticas, técnicas y creencias que guían el proceso de desarrollo de un programa
- Existen diferentes paradigmas:
  - **Imperativo**: un programa se compone de instrucciones que “dan” órdenes a la computadora
  - **Declarativo**: el programa describe los “resultados esperados”, sin listar los pasos a llevar a cabo
  - **Lógico**: el programa se desarrolla con enunciados de lógica de primer orden [\[más información\]](#)
  - **Funcional** (programación estructurada): los programas se descomponen en funciones
    - Cada función tiene una serie de datos de entrada y otros de salida
  - **Orientado a objetos**: se representan objetos (entidades que representan elementos del problema)

# 6. Programación estructurada

## La programación estructurada

- Emplean lenguajes de programación procedimentales
- Cada sentencia indica a la computadora una tarea concreta (sumar dos números...)
- Los programas más grandes se descomponen en funciones
- Cada función tiene un propósito determinado y resuelve una tarea concreta
- Las funciones se agrupan en módulos o ficheros
- Un programa se compone de uno o muchos ficheros

# 7. Programación orientada a objetos

## La programación orientada a objetos

- Es el paradigma de programación más utilizado
- Permiten representar los diferentes elementos u **objetos** del problema
- Los objetos del mundo real tienen **atributos** y **comportamiento**:
  - **Atributos**: son las características de los objetos (Coche: matricula, color, modelo...)
  - **Comportamiento**: representan las acciones que los objetos pueden llevar a cabo (Coche: frenar...)

# 8. Tipos de software

## Software, software del sistema, software de aplicación y sistema operativo

- El **software** es un conjunto de instrucciones que gobiernan el **hardware** de la computadora
- Por su parte, el **software del sistema** gestiona los recursos de la computadora
- El **software de aplicación** permite al usuario de una computadora realizar tareas concretas:
  - Crear documentos de ofimática
  - Conectarse a Internet...
- El **sistema operativo** (OS) es el software más importante del sistema:
  - Controla y gestiona los recursos de la computadora
  - Existen multitud de sistemas operativos: UNIX, Linux, Windows, MacOS, Android, iOS...



# 9. Lenguajes de programación

## Concepto de lenguaje de programación

- El lenguaje de programación permite que un algoritmo se **ejecute** en una computadora
- Puede ser **traducido** para que lo entienda el hardware de la computadora (código máquina)
- Existen multitud de lenguajes: BASIC, C, C++, C#, Java, Visual Basic, PHP, JavaScript...
- Clasificación de los lenguajes:
  - **Lenguaje de alto nivel**: mayor nivel de abstracción, pero son más complejos de usar
  - **Lenguaje de bajo nivel**: más fáciles de usar, pero dependen mucho del hardware de la computadora
  - **Lenguaje máquina** (código máquina): lo entiende directamente la máquina (0 o 1)

# 9. Lenguajes de programación

## Lenguaje interpretado vs. lenguaje compilado

- Un **lenguaje interpretado** utiliza un intérprete:
  - El intérprete toma el código fuente, lo interpreta y lo ejecuta (independiente de la máquina)
- Un **lenguaje compilado** utiliza un compilador:
  - El compilador traduce el programa
  - El programa se traduce en un programa ejecutable
  - El programa ejecutable puede ser comprendido por el hardware de la máquina
  - La traducción o compilación es dependiente del hardware de la máquina

# 9. Lenguajes de programación

## Fases de la compilación de un programa

### 1. Escritura del **programa fuente**

- El programa se escribe usando un editor o un entorno de desarrollo (IDE)

### 2. Compilación del programa

- Se emplea el **compilador** o programa traductor

### 3. Obtención del programa objeto (objeto binario)

### 4. Enlazado con las **librerías** (linker): obtiene el programa ejecutable

### 5. Ejecución del **programa ejecutable** (escrito en lenguaje máquina)

# 9. Lenguajes de programación

## Historia de C++

- Nace a partir de un estándar ANSI/ISO, plenamente estandarizado desde 1998
- Está basado en C (lenguaje de programación de sistemas por excelencia)
- Supone una extensión de C hacia el paradigma de la **programación orientada a objetos**
- Casi todas las sentencias de C tienen una sentencia correcta en C++ (no a la inversa)

# 9. Lenguajes de programación

## Características de C++

- Lenguaje orientado a objetos (abstracción, encapsulado, herencia y polimorfismo)
- Estándar ISO (ANSI C++) [\[más información\]](#)
- Posee una biblioteca estándar de funciones
- Permite el control de excepciones
- Compatibilidad con el lenguaje C

# Créditos de las imágenes y figuras

## Cliparts e iconos

- **Obtenidos mediante la herramienta web [IconFinder](#)** (según sus disposiciones):
  - Diapositiva 1
  - Según la plataforma IconFinder, dicho material puede usarse libremente (free comercial use)
  - A fecha de edición de este material, todos los cliparts son free for comercial use (sin restricciones)

## Resto de diagramas y gráficas

- Se han desarrollado en PowerPoint y se han incrustado en esta presentación
- Todos estos materiales se han desarrollado por el autor
  - Si se ha empleado algún icono externo, este se rige según lo expresado anteriormente